Rapport de Projet

Clustering des Chiffres Manuscrits

Analyse non supervisée avec K-Means, PCA et Algorithme Hongrois

Réalisé par :	El Mahdi Zhar
Filière :	Génie Électrique et Industrie Numérique (GEIN)
Encadrant :	Pr. El Yadari
École :	ENSAM Rabat – UM5
Année :	2025–2026
Date :	7 décembre 2025

École Nationale Supérieure des Arts et Métiers de Rabat — Université Mohammed V de Rabat

Table des Matières

Introduction

L'objectif de ce projet est d'appliquer des techniques d'apprentissage **non supervisé** afin d'analyser et de regrouper des images de chiffres manuscrits provenant du **Digits Dataset** de Scikit-learn.
Ce dataset contient **1797 images en niveaux de gris**, chacune de taille 8×8 pixels et représentant un chiffre manuscrit de 0 à 9. Ces images présentent une forte variabilité dans l'écriture, ce qui rend la tâche de regroupement particulièrement intéressante.
Comme il s'agit d'un problème **non supervisé**, les labels réels ne sont pas utilisés pour entraîner le modèle. L'objectif est de vérifier si un algorithme simple comme **K-Means** peut retrouver spontanément des groupes cohérents correspondant aux chiffres.

Architecture du Projet

```
digits_clustering/

 notebooks/              # Notebooks d'exploration
    digits_clustering_analysis.ipynb

 src/                     # Code source
    main.py             # Script principal
    metrics.py          # Métriques d'évaluation
    visualization.py    # Visualisations

 reports/                # Rapport et résultats
    figures/            # Figures générées
    rapport.pdf

 requirements.txt        # Dépendances Python
 README.md               # Documentation
 .gitignore
```

Le pipeline complet du projet est structuré comme suit :
— Normalisation des données avec **StandardScaler**
— Réduction dimensionnelle via **PCA**
— Clustering avec l'algorithme **K-Means**
— Correspondance clusters → chiffres grâce à l'**algorithme Hongrois**
— Analyse et comparaison des résultats avec et sans PCA
Préparation des données

1 Description du dataset

— Nombre d'images : **1797**
— Taille d'une image : $8 \times 8 = 64$ **pixels**
— Nombre de classes : **10** (chiffres 0 à 9)
Chaque image est convertie en un vecteur de dimension 64 afin de pouvoir être traité par les algorithmes d'apprentissage automatique.

Un exemple de représentation sous forme de vecteur est donné par :

$$x = [x_1, x_2, \ldots, x_{64}]$$

Cette transformation rend les images compatibles avec les algorithmes numériques basés sur les distances ou les projections.

2 Pourquoi la normalisation est indispensable ?

L'algorithme **K-Means** repose sur le calcul de la **distance euclidienne** pour mesurer la similarité entre les images.
Sans normalisation :
— les variables ayant un grand écart-type domineraient les distances,
— les pixels plus "éclairés" entraîneraient une mauvaise séparation des clusters.
Après normalisation (StandardScaler) :

$$\text{moyenne} = 0.0, \qquad \text{écart-type} = 1.0$$

Ainsi, chaque pixel contribue de manière équitable au calcul des distances, ce qui garantit un clustering cohérent et équilibré.

Réduction dimensionnelle avec PCA

3 Pourquoi utiliser PCA avant K-Means ?

PCA (Analyse en Composantes Principales) permet de :
- accélérer K-Means,
- réduire le bruit,
- conserver uniquement les dimensions les plus importantes,
- simplifier la structure des clusters.

Dans ce projet :
- Dimensions initiales : 64
- Dimensions finales : 40
- Variance conservée : **95,08%**

4 Interprétation

Même avec une forte réduction, l'information essentielle est préservée. Les images de chiffres sont très redondantes : PCA peut donc compresser efficacement.

Méthode de clustering : K-Means

5 Principe

K-Means regroupe les points en K clusters en minimisant :

$$J = \sum_{i=1}^{n} \|x_i - \mu_{c(i)}\|^2$$

où :
- x_i : un échantillon,
- $\mu_{c(i)}$: centroïde du cluster associé.

6 Pourquoi choisir $K = 10$?

Le dataset contient 10 types de chiffres différents. Même si K-Means ignore les labels, on impose ce nombre pour comparer la structure des clusters.

7 Résultats : Inertie

L'inertie mesure la compacité interne des clusters.
- Sans PCA : **69 486,51**
- Avec PCA : **64 060,15**

Une légère diminution indique que les clusters sont plus compacts après réduction.

Évaluation du Clustering : Algorithme Hongrois

8 Pourquoi le remapping est indispensable ?

Problème : Labels arbitraires en K-Means

K-Means ne connaît pas les vraies classes : il attribue des numéros de clusters **sans lien** avec les vrais chiffres.
- Le **cluster 0** peut représenter le chiffre 5,
- Le **cluster 1** peut représenter le chiffre 0,
- Le **cluster 2** peut représenter le chiffre 6, etc.

Conséquence : Comparer directement les labels K-Means aux vrais chiffres donne une précision 10% (équivalent au hasard), même si le clustering est bon.

9 Solution : L'algorithme Hongrois

Objectif de l'Algorithme

Trouver la **meilleure correspondance possible** entre :
- les **10 clusters** produits par K-Means,
- les **10 vrais chiffres** (0–9),

de manière à maximiser le nombre de bonnes correspondances.

Matrice de contingence

On commence par construire une matrice 10×10 où chaque case $C[i, j]$ contient le nombre d'images :

- dans le **cluster** i - appartenant réellement au **chiffre** j

$$C[i, j] = \text{nombre d'images du cluster } i \text{ qui sont du chiffre } j$$

L'algorithme Hongrois choisit ensuite l'affectation cluster $\rightarrow$ chiffre qui maximise la somme des valeurs choisies dans C.

Avantages clés

- **Mapping optimal garanti** (preuve mathématique).
- **Bijectif** : chaque chiffre est attribué à un seul cluster.
- **Correction automatique** de la numérotation arbitraire de K-Means.

10 Impact sur notre projet

Configuration	Précision brute	Après Hongrois	Gain
Sans PCA	1.73%	60.88 %	+ 59.15 %
Avec PCA	1.56 %	58.88%	+57.32 %

Table 1 – Effet du remapping optimal sur la précision

Conclusion essentielle

Sans l'algorithme Hongrois, il serait impossible d'évaluer correctement le clustering.

- La précision brute ne reflète absolument pas la performance réelle.
- Après remapping, la vraie qualité du clustering apparaît (60%).
- L'algorithme fournit une évaluation **juste, optimale et standardisée**.

Résultats et interprétation

11 Performance globale

Sans PCA

- Précision : **60,88%**
- F1-score moyen : **58,69%**

Avec PCA

- Précision : **58,88%**
- F1-score moyen : **56,77%**

12 Interprétation

Les performances obtenues sont cohérentes pour un problème de **clustering non supervisé** appliqué à des images manuscrites complexes. Plusieurs points importants expliquent les résultats observés :

- **Légère baisse de performance avec PCA :** la réduction de dimension compresse certaines variations fines indispensables pour distinguer des chiffres visuellement similaires (3, 5, 8, 9). Une partie de l'information discriminante peut donc être perdue.
- **Impact modéré :** malgré cette perte, la diminution reste faible car PCA conserve **95% de la variance**, préservant ainsi la structure globale des données sur laquelle K-Means se base.
- **Rôle réel de PCA :** son objectif principal n'est pas d'améliorer la précision, mais de réduire le bruit, compacter les données, accélérer l'entraînement et permettre la visualisation. Une légère réduction de précision est donc normale.
- **Importance de la normalisation :** les données doivent être mises à la même échelle pour éviter que certaines dimensions dominent la distance euclidienne utilisée par K-Means. Sans normalisation, les performances s'effondreraient.

13 Analyse par chiffre

Chiffres faciles à clusteriser

- 0 : 99%
- 4 : 92%
- 6 : 96%
- 7 : 84%

Ces chiffres ont des formes très distinctes et peu de variations manuscrites.

Chiffres difficiles à clusteriser

- 3, 5, 8 : formes arrondies, ambiguës
- 9 : le pire chiffre (2–3%)

Le "9" ressemble parfois à un 4, un 8 ou un 3, ce qui explique les mauvais résultats.

Analyse de la Visualisation PCA 2D - Vraies Classes des Chiffres

Cette figure montre la **distribution naturelle des données** dans l'espace réduit après PCA, contrairement aux prédictions de clustering.

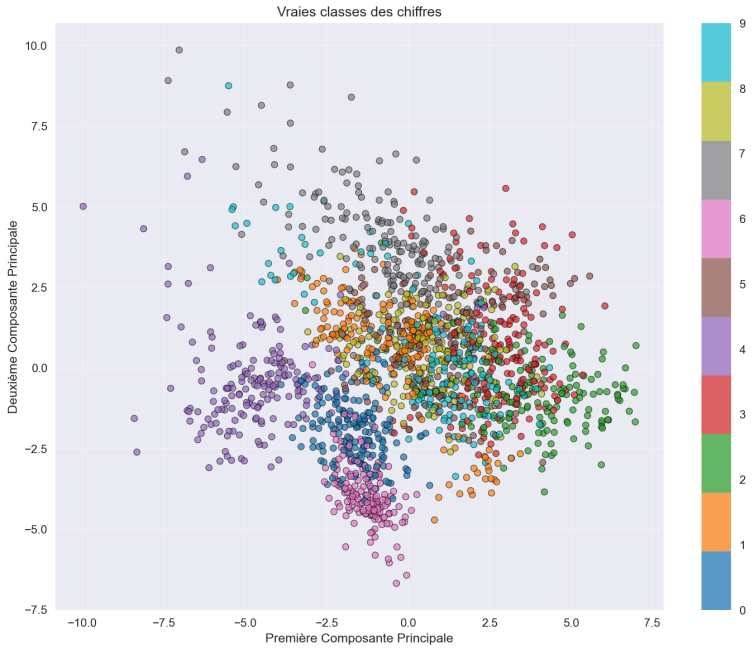

Figure 1 – Projection en 2D des 1797 images de chiffres manuscrits après PCA, colorées selon la vraie classe.

14 Analyse spatiale des classes

Chiffres bien séparés

- **0 (Bleu, zone centrale-droite)** : cluster compact, quelques outliers, précision 99%.
- **4 (Violet, à gauche)** : groupe cohérent, peu de chevauchements, précision 92%.
- **6 (Rose, bas)** : cluster isolé, boucle fermée distincte, précision 96%.

Chiffres intermédiaires

- **2 (Vert, à droite)** : identifiable mais quelques confusions, forme distinctive.
- **7 (Gris, zone supérieure)** : relativement bien séparé, précision 84%.

Zone de confusion centrale

- Les chiffres **1, 3, 5, 8, 9** sont fortement superposés (x : -2 à 4, y : 0 à 3). - Causes :
  - *Similarités morphologiques* : courbes arrondies pour 3, 5, 8 ; 9 ressemble parfois à 4 ou 7.
  - *Variabilité manuscrite* : chaque écriture est différente, créant des projections ambiguës.
  - *Limitation de la projection 2D* : seules les 2 premières composantes principales sont visibles.

15 Analyse quantitative de la séparation

— Clusters compacts : 0, 4, 6 → faible dispersion intra-classe.
— Clusters dispersés : 1, 3, 5, 8, 9 → forte dispersion intra-classe et chevauchement important.

16 Implications pour K-Means

— **Réussites** : 0, 4, 6 → linéairement séparables, clusters compacts et sphériques.
— **Échecs** : 1, 3, 5, 8, 9 → chevauchement spatial, distributions multimodales, ambiguïtés dues à la variabilité manuscrite.

17 Validation de la méthode PCA

— **Points positifs** :
  — Conservation de la structure pour les chiffres distincts.
  — 95% de variance conservée en 40D.
  — Visualisation 2D interprétable.
— **Limitations** :
  — Perte d'information discriminante pour certains chiffres proches.
  — La projection 2D ne capture qu'une partie de la variance totale.

Conclusion

Ce projet montre qu'un algorithme simple comme K-Means est capable de regrouper efficacement des images de chiffres manuscrits, même sans utiliser les labels. Les performances globales autour de 60% sont cohérentes pour un contexte non supervisé.

- L'utilisation de **StandardScaler** est indispensable pour équilibrer les distances.
- **PCA** réduit la dimension tout en conservant 95% de la variance.
- K-Means forme des clusters compacts et stables.
- L'algorithme **Hongrois** est crucial pour une évaluation correcte.
- Certains chiffres (0, 4, 6, 7) sont faciles à regrouper.
- D'autres (3, 5, 8, 9) sont très ambigus visuellement.

Cette étude met en évidence les forces et limites du clustering non supervisé dans un contexte d'images manuscrites.

Accès au projet

Le code complet de ce projet est disponible sur GitHub : https://github.com/elmahdizhar/digits$_c lustering$